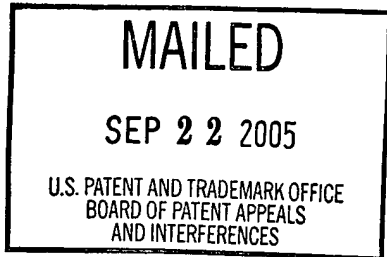The opinion in support of the decision being entered today was <u>not</u> written
for publication and is <u>not</u> binding precedent of the Board.

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES

<u>Ex parte</u> K. RUSTAN M. LEINO, TODD DAVID MILLSTEIN
and JAMES B. SAXE

Appeal No. 2005-1860
Application No. 09/754,890[1]

ON BRIEF

Before JERRY SMITH, KRASS and SAADAT, <u>Administrative Patent Judges</u>.
SAADAT, <u>Administrative Patent Judge</u>.

<u>DECISION ON APPEAL</u>

This is a decision on appeal from the Examiner's final

rejection of claims 1-47, which are all of the claims pending in

this application.

We reverse.

<u>BACKGROUND</u>

Appellants' invention is directed to computer verification

systems that automatically verify the correctness of a computer

program with respect to predefined criteria. By inserting

---

[1] Application for patent filed January 5, 2001, which claims the filing
priority benefit under 35 U.S.C. § 119 of the provisional Application No.
60/174,952, filed January 7, 2000.

program flow control labels into sub-equations of a logical

equation called a verification condition, conditional branch

points in a program may be identified (specification, pages 3 &

4).

Representative independent claim 1 is reproduced below:

1. A method of verifying with static checking whether a
specified computer program satisfies a predefined set of
conditions, comprising:

converting the program into a logical equation representing
the predefined set of conditions as applied to instructions and
elements of the instructions of the program, the converting step
including inserting flow control labels into the sub-equations of
the logical equation, the flow control labels identifying
conditional branch points in the specified computer program;

applying a theorem prover to the logical equation to
determine the truth of the logical equation, and when the truth
of the logical equation cannot be proved, generating at least one
counter-example identifying one of the conditions, one or more
variable values inconsistent with the one condition, and any of
the flow control labels for conditional branch points of the
program associated with the identified variable values; and

converting the at least one counter-example into an error
message that includes a program trace that identifies a path
through the computer program when the counter-example identifies
one or more of the flow control labels.

The Examiner relies on the following references:

Chan et al. (Chan)   4,920,538   Apr. 24, 1990

Rickel et al. (Rickel)   5,854,924   Dec. 29, 1998

David L. Detlefs et al. (Detlefs), "Extended Static Checking,"
Compaq Systems Research Center, SRC Research report #159
(December 18, 1998), pp. 1-44.

Claims 1-5, 8, 11, 14, 17, 20, 22-27, 30, 33, 36, 39, 42, 45 and 47 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Detlefs and Chan.

Claims 6, 7, 9, 10, 12, 13, 15, 16, 18, 19, 21, 28, 29, 31, 32, 34, 35, 37, 38, 40, 41, 43, 44 and 46 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Detlefs, Chan and Rickel.

Rather than reiterate the opposing arguments, reference is made to the briefs and answer for the respective positions of Appellants and the Examiner. Only those arguments actually made by Appellants have been considered in this decision. Arguments which Appellants could have made but chose not to make in the brief have not been considered (37 CFR § 41.37(c)(1)(vii)).

OPINION

The Examiner relies on Detlefs for teaching a method of verifying with static checking that converts the program into a logical equation except for the claimed step of inserting flow control labels (answer, page 4). The Examiner further relies on Chan for teaching markers and scalars as the claimed "flow control labels" for identifying conditional branch points and takes the position that Detlefs' counter example traces the source of the error while Chan's markers and scalars identify paths of execution (answer, page 6). In order to justify the

combination, the Examiner equates the "error message" of Detlefs
and the markers of Chan with the claimed identifying the
positions and the paths through the program based on the inserted
flow control labels (answer, page 21).

Appellants argue that what the Examiner relies on Detlefs is
actually a "checker to report specific error messages" instead of
teaching an error message that includes a program trace that
identifies a path through the computer program, as recited in
claim 1 (brief, page 5; reply brief, page 2). Additionally,
Appellants assert that Chan is merely related to a self-verifying
controller that verifies valid branches along the execution path
after they are detected and isolated (brief, the last paragraph
of page 7). Appellants further assert that the combination of
Detlefs and Chan, instead of the claimed using flow control
labels inserted in the sub-equations and identifying conditional
branch points, provides for identifying the location of a
specific error type (Detlefs) or computing a marker at runtime to
detect wild branches (Chan) (brief, page 7).

In rejecting claims under 35 U.S.C. § 103, the Examiner
bears the initial burden of presenting a prima facie case of
obviousness. See In re Rijckaert, 9 F.3d 1531, 1532, 28 USPQ2d
1955, 1956 (Fed. Cir. 1993). To reach a conclusion of
obviousness under § 103, the examiner must produce a factual

basis supported by teaching in a prior art reference or shown to
be common knowledge of unquestionable demonstration. Such
evidence is required in order to establish a <u>prima facie</u> case.
<u>In re Piasecki</u>, 745 F.2d 1468, 1471-72, 223 USPQ 785, 787-88
(Fed. Cir. 1984). The Examiner must not only identify the
elements in the prior art, but also show "some objective teaching
in the prior art or that knowledge generally available to one of
ordinary skill in the art would lead the individual to combine
the relevant teachings of the references." <u>In re Fine</u>, 837 F.2d
1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

Upon a review of the applied prior art, we find ourselves
unpersuaded by the Examiner's reasoning that the static checker
of Detlefs provides for other than reporting specific error
messages without allowing the error message to include a program
trace for identifying the path through the program.
Particularly, Detlefs labels the subformulas of the theorem
prover to track specific error source and type (page 29, section
6, 2nd paragraph). As pointed out by Appellants (brief, page 6),
Chan also fails to provides for the missing flow control labels
that identify a path through the computer program. Chan, at
best, discloses markers for verifying paths of execution by
comparing them with permissible execution paths in order to avoid
wild paths (col. 1, lines 38-53). Chan performs the verifying

task by computing a marker at runtime and comparing it with a stored marker to detect wild branches (col. 1, lines 41-43) and generates a global label path identifier for a permissible path which computes a check marker to be matched with other paths (col. 1, lines 43-49).

Thus, although we agree with the Examiner that some kind of path identifier is recognized by Chan for detecting and isolating invalid branches, we do not find any specific teaching in the reference that relates to the claimed program trace that identifies a path through the computer program. In concluding that Chan's markers could also identify paths of execution in a computer program, the Examiner attempts to forge a combination of a labeled error messages that identify specific errors and markers that identify wild branches. Thus, assuming, arguendo, that it would have been obvious to combine Detlefs with Chan, as held by the Examiner, the combination would still fall short of teaching or suggesting the claimed feature of including a program trace that identifies a path through the computer program. Accordingly, as the Examiner has failed to set forth a prima facie case of obviousness, we do not sustain the 35 U.S.C. § 103 rejection of claims 1-5, 8, 11, 14, 17, 20, 22-27, 30, 33, 36, 39, 42, 45 and 47 over Detlefs and Chan.

We note that the Examiner, in rejecting the remaining claims, in addition to Detlefs and Chan, further relies on Rickel which neither includes any teachings that read on the disputed claimed features nor provides any suggestion for combining the references to overcome the deficiencies of Detlefs and Chan as discussed above. Accordingly, we do not sustain the 35 U.S.C. § 103 rejections of claims 6, 7, 9, 10, 12, 13, 15, 16, 18, 19, 21, 28, 29, 31, 32, 34, 35, 37, 38, 40, 41, 43, 44 and 46 over Detlefs, Chan and Rickel.

CONCLUSION

In view of the foregoing, the decision of the Examiner

rejecting claims 1-47 under 35 U.S.C. § 103 is reversed.

REVERSED

JERRY SMITH   )
Administrative Patent Judge   )
)
)
)
) BOARD OF PATENT
ERROL A. KRASS   )   APPEALS
Administrative Patent Judge   )   AND
)   INTERFERENCES
)
)
)
MAHSHID D. SAADAT   )
Administrative Patent Judge   )

MDS/ki

8

Appeal No. 2005-1860
Application No. 09/754,890

Hewlett Packard Company
P.O. Box 272400
3404 E. Harmony Road
Intellectual Property Administration
Fort Collins, CO 80527-2400